AD 748 207

PERSONNEL-ASSIGNMENT BY MULTIOBJECTIVE PROGRAMMING

BY

M. A. POLLATSCHEK

TECHNICAL REPORT 72-13
JULY 1972

OPERATIONS
RESEARCH
HOUSE

Stanford
University
CALIFORNIA

PERSONNEL-ASSIGNMENT BY MULTIOBJECTIVE PROGRAMMING

by

M. A. Pollatschek

TECHNICAL REPORT 72-13

July 1972

DEPARTMENT OF OPERATIONS RESEARCH

Stanford University
Stanford, California

Research and reproduction of this report was partially supported by the Office of Naval Research under contract N-00014-67-A-0112-0011; U.S. Atomic Energy Commission Contract AT(04-3)-326-PA #18; and National Science Foundation, Grant GP 31393.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Department of Operations Research Stanford University	2a. REPORT SECURITY CLASSIFICATION 2b. GROUP

3. REPORT TITLE

PERSONNEL-ASSIGNMENT BY MULTIOBJECTIVE PROGRAMMING

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report

5. AUTHOR(S) (Last name, first name, initial)

POLLATSCHEK, M. A.

6. REPORT DATE July 1972	7a. TOTAL NO. OF PAGES 23	7b. NO. OF REFS 11

8a. CONTRACT OR GRANT NO. N00014-67-A-0112-0011 b. PROJECT NO. NR-047-064 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 72-13 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. AVAILABILITY/LIMITATION NOTICES

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Operations Research Program Code 434 Office of Naval Research Washington, D.C. 20360

13. ABSTRACT

Assigning personnel is a task where several goals should be simultaneously achieved. It is demonstrated that multiobjective programming in combinatoric medium is possible both methodologically and computationally. Following Geoffrion, it is assumed that the decision-maker has in mind an implicity function mapping the numerical values of the objectives to the real line. The following strategy is suggested: in each stage of a branch-and-bound type procedure the DM is requested to choose the first- and second-best assignment from a given set, and compare several pairs of assignments. The computation is reducible to well-known problems of assignment and shortest routes for which efficient solution-techniques exist.

DD FORM 1 JAN 64 1473

Unclassified

Security Classification

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
ASSIGNMENT						
MULTIOBJECTIVE PROGRAMMING						
BRANCH-AND-BOUND PROCEDURE						
SHORTEST ROUTE PROBLEM						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
____ ."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
____ ."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
____ ."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.

DD FORM 1 JAN 64 1473 (BACK)

Unclassified

Security Classification

PERSONNEL-ASSIGNMENT BY MULTIOBJECTIVE PROGRAMMING

M. A. Pollatschek
Stanford University

Abstract

Assigning personnel is a task where several goals should be simultaneously achieved. It is demonstrated that multiobjective programming in combinatoric medium is possible both methodologically and computationally. Following Geoffrion, it is assumed that the decision-maker has in mind an implicit function mapping the numerical values of the objectives to the real line. The following strategy is suggested: in each stage of a branch-and-bound type procedure the DM is requested to choose the first- and second-best assignment from a given set, and compare several pairs of assignments. The computation is reducible to well-known problems of assignment and shortest routes for which efficient solution-techniques exist.

Key-words: Assignment
Multiobjective Programming
Branch-and-Bound Procedure
Shortest Route Problem

INTRODUCTION

Kuhn's Hungarian assignment algorithm should be in principle a good tool for personnel-assignment to jobs, however its application is reportedly scarce [9]. A successful application has recently been reported [4] in which servicemen were assigned to several posts and the "cost" of an assignment was the dollar costs of moving the individual, his family and belongings to a new place of residence. However, not each assignment was acceptable since certain policy-rules had to be considered as well.

This example pinpoints the difficulty in the application of the algorithm: namely, the policy-rules cannot be casted into single scalars attached to the assignments the sum of which is to be minimized. One general approach to circumvent this fact is to attach a vector to the assignment and thus to solve a multiple objective-function problem. (For a summary see [11].) If this is possible we still have to face a further difficulty: namely, the definition of the optimum. Clearly, no solution optimizes all the objective functions simultaneously and no apparently simple function can weigh properly the several objectives. Geoffrion [6] proposed that such function exists in the case of convex multiobjective function and its derivatives at given points can be assessed by the decision-maker (DM) although we are unable to construct the function explicitly. Hence, what is needed in such cases is an algorithm influenced properly by the DM. The DM continuously exercises his judgement during the computational framework by choosing between the alternatives.

The main objective of the paper is to demonstrate that multi-objective programming in combinatoric medium is possible both methodologically and computationally and in this sense the presented material seems to be novel. On the other hand most of the techniques are taken from well-established fields as branch-and-bound procedures [7], [8] and network analysis [3], [5] which the reader is assumed to be familiar with.

METHODOLOGY: FORMULATION AND STRATEGY OF SOLUTION

Assume that there are n candidates and m jobs, $m \leqq n$, and there are K objectives, $K \geqq 1$. The deviation caused by assigning candidate i to job j from the ideal value, 0, concerning objective k is $c_{ij}^k \geqq 0$, $k = 1, 2, \ldots, K$. Define $x_{ij} = 1$ if candidate i is assigned to job j and $x_{ij} = 0$ otherwise and let f^k be:

$$f^k = \sum_{i,j} c_{ij} \; x_{ij}.$$

f^k measures the distance of an assignment pattern $X = \{x_{ij}\}$ from the ideal value, 0.

F is not known explictly. However, it is an increasing function of f_k, $k = 1, 2, \ldots, K$ if the remaining variables are held constant.*

The decision-maker wants to minimize F.

If F is independent of X this approach is equivalent with that proposed by Geoffrion in the continuous, convex-programming case [6]. Thus, it can be considered as generalization of [6] in a discrete problem.

To illustrate, let us discuss assigning servicemen to posts when there are three considerations:

(i) Cost caused by moving the individual, his family and belongings to the new post. The total cost of all the

---

*It is interesting to observe that if $F = \sum c_{ij} \, x_{ij} + M(1 - \det(I - X))$ for $n = m$, and $M \to \infty$ we obtain a correct formulation for the notorious traveling salesman problem [10]. This shows the generality of this approach.

assignment should be kept low. Define c^1_{ij} as this cost in $\$$.

(ii) Reliability of the system after assignment. Let p_{ij} the estimated probability of error made by servicemen i in post j. Postulating independency, the probability of no error in the whole system is $\pi(1 - p_{ij})$ where the product is carried over (i, j) such that $x_{ij} = 1$. This should be kept reasonably high. Therefore define $c^2_{ij} = - \log (1 - p_{ij})$.

(iii) Certain pattern of assignments are not feasible due to, say, union-contracts which forbid them.

Function F in this case weigh the desirability of low cost against the advantage of high reliability and also avoids the forbidden patterns.

F is actually a map from the k-tuple $(f^1, f^2, \ldots, f^k)$ to $y \epsilon R^1$: $y = F(f^1, f^2, \ldots, f^k)$. Hence if one has p k-tuples $(f^1_i, f^2_i, \ldots, f^k_i)$ $i = 1, 2, \ldots, p$, and the corresponding y_i, the DM should be able the point out the k-tuple leading to the first or second minimum of y or to compare the distances between two pairs of y_i, i.e., if $|y_1 - y_2| < |y_3 - y_4|$, etc. Therefore it is sensible to ask him about his first or second preference or compare the distances between given pairs of k-tuples. This may prove incorrect in practice. In order to compensate for this strong assumption the DM will be allowed to change his mind retroactively during the execution of the algorithm, mainly as the results of the earlier decisions concerning preferences turn out at later

stages. This necessiates a backtrack-type algorithm similar to that given in [8].

The DM assigns the m jobs to m candidates one at a time.

In assignment of ℓ given jobs to ℓ given candidates, $\ell = 0, 1, \ldots, m$ is termed an ℓ-assignment (ℓ - a). Let ${}_{\ell}\ell f^k$ be the cost of the specific ℓ - a plus the cost of optimal assignment of the remaining jobs and candidates concerning objective k only. Clearly, ${}_{\ell}\ell f^k$ is a lower bound of f^k in an ℓ - a. If (i, j) is to be the following assignment from the so far unassigned jobs and candidates, it can be shown (see the following section) that ${}_{\ell+1}\ell f^k - {}_{\ell}f^k$ is a function of the specific ℓ - a, C^k and (i, j) denoted by ${}_{\ell}\Delta f^k_{i,j}$. ${}_{\ell}\ell f^k$, and ${}_{\ell}\Delta f^k_{ij}$ are displayed to the DM. On the basis of this information, the DM indicates his first and second preferences for the remaining candidates for each remaining job. These will be denoted by $i_1(j)$ and $i_2(j)$ for job j. As a following step the DM is requested to indicate the job j for which the difference between $\{{}_{\ell}\ell f^k + {}_{\ell}\Delta f^k_{i_1(j),j} \mid k = 1, 2, \ldots, K\}$ and $\{{}_{\ell}\ell f^k + {}_{\ell}\Delta f^k_{i_2(j),j} \mid k = 1, 2, \ldots, K\}$ is the greatest. This job will be denoted by $j(\ell)$. Following the beuristics in [7] and [8] the branching step is performed on $j(\ell)$. Backtracking takes place only if the DM so desires and he specifies also the node in the branch-and-bound tree toward which the algorithm backtracks. Like in [8], after backtracking, the proper $c_{i_1(j(\ell)),j(\ell)}$ is set to a high number, and the calculation is restarted from that point.

CALCULATIONS

For each ℓ - a, $\ell = 0, 1, \ldots, m - 1$ there are $(m - \ell)$ remaining jobs and $(n - \ell)$ remaining candidates. The corresponding $K(n - \ell) \times (m - \ell)$ assignment-problem are solved independently for the K objectives by standard methods [2], [7]. Let v^k be the optimal value of the k-th objective function. Then the lower bounds on f^k is given by

$$v^k + \sum_{p=1}^{\ell} c_{i_1(j(p)),j(p)},$$

where if $\ell = 0$, the sum is defined as 0.

For the calculation of ${}_{\ell}\Delta f^k_{ij}$ we need some further definitions. Without losing generality we proceed by assuming $\ell = 0$, and observing only one (arbitrary) objective. Therefore indeces k and ℓ will be omitted in the foregoing.

Associate a bipartite, directed graph to an arbitrary but fixed optimal assignment with vertex-sets $\epsilon_1 = \{1, 2, \ldots, n\}$ and $\epsilon_2 = \{1, 2,$ $\{1, 2, \ldots, m\}$ corresponding to the row and column indices of matrix C whose elements are c_{ij}. Let the optimal pairs (i,j) correspond to edges (i,j), $i \epsilon \epsilon_1$, $j \epsilon \epsilon_2$ and all the other pairs (i,j) correspond to edges (j,i), $i \epsilon \epsilon_1$, $j \epsilon \epsilon_2$. Reindex the rows so that the optimal pairs c_{ij} are on the diagonal of C. This is achieved by premultiplying C with a suitable permutation-matrix, P. Thus, the optimal paris of PC are (j,j), $j + 1, 2, \ldots, m$. In the following we assume this index-system: thus c_{ij} indicates the (i,j)-element in PC. In assignment, $\mathcal{A}$ is a matching in this bipartite graph.

The compressed graph is defined as follows: it has a vertex-set $\epsilon = \{1, 2, \ldots, n\}$ and for each arc (j,i) $i \epsilon \epsilon_1$, $j \epsilon \epsilon_2$ in the bipartite graph we have an edge (i,j) in the compressed graph with "distances" or weights $w_{ij} = c_{ij} - c_{jj}$. Additionally there are artificial arcs (i,j) with weights $w_{ij} = 0$ for $i \epsilon \{1, 2, \ldots, m\}$, $j \epsilon \{m+1, m+2, \ldots, n\}$ and $w_{ij} = \infty$ for $i \epsilon \{m+1, m+2, \ldots, n\}$, $j \epsilon \{m+1, m+2, \ldots, n\}$ and $i \neq j$. $w_{ii} = 0$ for each $i + 1, 2, \ldots, n$ by definition.

Let $\mathcal{C}$ be a family of disjoint circuit in the compressed graph. A family of disjoint circuits in the bipartite graph containing the corresponding arcs will be denoted also by $\mathcal{C}$. Since there exists a one-to-one correspondence between the families this abuse is well-defined.

<u>Lemma</u>. Let $\mathcal{A}_1$ be an optimal assignment and $\mathcal{C}$ - a family of disjoint circuits in the compressed graph. Then an assignment $\mathcal{A}_2$ may be obtained from $\mathcal{A}_1$ and $\mathcal{C}$ in the following manner: if $\mathcal{C}$ contains a nonartificial arc (i,j) then pair (j,j) is replaced by pair (i,j). Moreover, each $\mathcal{A}_2 \neq \mathcal{A}_1$ can be obtained from some $\mathcal{C}$ by this contruction.

<u>Proof</u>. Take any $\mathcal{A}_2 \neq \mathcal{A}_1$. Then consider the set of arcs $\mathcal{A}_2 \cup \mathcal{A}_1 - (\mathcal{A}_2 \cap \mathcal{A}_1)$ in the bipartite graph. They may form either circuits or pathes from $i \epsilon \{m+1, m+2, \ldots, n\}$ to $j - \{1, 2, \ldots, m\}$ in the bipartite graph, and these circuit(s) and path(s) are disjoint. Therefore, the corresponding arcs in the compressed graph together with appropriate artifical arcs form a family of disjoint circuits, $\mathcal{C}$ proving the second part of the lemma. By reversing the arguments one verifies the first part.

The value of the shortest route from vertex p to g, denoted by w^*_{pg}, in the compressed graph is the minimum of Σw_{ij} where the sum is taken on a path from vertex p to vertex g.

The matrix of shortest routes, W^* with elements w^*_{ij} can be easily calculated by the Floyd-algorithm [1], [5].

Denote by Δf an $n \times n$ matrix whose elements in the first m rows are Δf_{ij}.

<u>Theorem</u>. $\Delta f = P^{-1}(W + W^{*t})$, where t denotes transposition.

<u>Proof</u>. Note that the (i,j)-elements $W^{(1)} + W^{*t}$ are the values of the shortest circuits of the compressed graph when (i,j) is constrained to be in these circuits. This follows from the fact that the shortest circuit containing arc (i,j) is composed from arc (i,j) and the shortest route from j to i and their values correspond to the appropriate elements of W and W^{*t} resp. Remark also that no negative circuits endanger the existence of W^{*t} despite the fact that W may have negative elements. To see this, by assuming negative circuit $\mathcal{C}$, we have

$$\sum_{(i,j)\in\mathcal{C}} w_{ij} = \sum_{(i,j)\in\mathcal{C}} c_{ij} - \sum_{j\in\mathcal{C}} c_{jj} < 0 \Rightarrow \sum_{(i,j)\in\mathcal{C}} c_{ij} < \sum_{j\in\mathcal{C}} c_{jj},$$

which implies the nonoptimality of the diagonal elements, c_{jj}, in contradiction to our original assumption.

By the construction of the lemma, we obtain from the optimal assignment $\mathcal{A}_1$, and $\mathcal{C}$ containing (r,s) a new assignment, $\mathcal{A}_2$, which contains (r,s). Let $v(\mathcal{A})$ be the sum of C_{ij} on the arcs of $\mathcal{A}$ in the bipartite graph. Then

$$v(\mathcal{A}_2) = v(\mathcal{A}_1) + v(\mathcal{C}) - v(\mathcal{C} \cap \mathcal{A}_1) = v(\mathcal{A}_1) + \sum_{(i,j)\in\mathcal{C}} w_{(i,j)}.$$

Hence,

$$\sum_{(i,j)\in\mathcal{C}} w_{i,j} = v(\mathcal{A}_2) - v(\mathcal{A}_1),$$

and by the one-to-one correspondence of $\mathcal{A}_2$ and $\mathcal{C}$ (for a fixed $\mathcal{A}_1$):

$$\underset{\mathcal{C}}{\mathrm{Min}} \left[\sum_{(i,j)\in\mathcal{C}}^{1} w_{ij}\right] = \underset{\mathcal{A}}{\mathrm{Min}} \left[v(\mathcal{A}_2) - v(\mathcal{A}_1)\right] = \Delta f_{r,s},$$

where both $\mathcal{C}$ and $\mathcal{A}_2$ contains arc (r,s). The left-hand side is the r,s element of $W^{(1)} + W^{*t}$, QED.

Let $\mathcal{A}$ be an optimal assignment concerning ${}_{\ell}C^k$ only (the subscript ℓ indicates that ${}_{\ell}C^k$ is a submatrix of C^k from which the rows and columns corresponding to ℓ - a are discarded) and $\mathcal{C}_{ij}$ the circuit corresponding to element (i,j) (and appropriate shortest route) in Δf. If in the $(\ell + 1)$ - a, the new assignment is (i,j) then an optimal assignment concerning ${}_{\ell+1}C^k$ is given by $\mathcal{A} \cup \mathcal{C}_{ij} - (\mathcal{A} \cap \mathcal{C}_{ij})$. Thus, by finding the solution to the shortest-route problem one is able to update the optimal assignment of ${}_{\ell+1}C^k$ from that of ${}_{\ell}C^k$.

Similar to ${}_{\ell+1}C^k$, ${}_{\ell+1}v^k$ can be also updated by the following formula:

$${}_{\ell+1}v^k = {}_{\ell}v^k + {}_{\ell}\Delta f^k_{i_1(j(\ell)),j(\ell)} - c^k_{i_1(j(\ell)),j(\ell)}.$$

ILLUSTRATIVE EXAMPLE

For illustrative purposes assume $K = 2$, $m = 6$, $n = 9$ and C^k are:

${}_oC^1 =$

	1	2	3	4	5	6
1	4	6	5	3	7	3
2	0	4	1	0*	3	2
3	4	1*	2	9	1	2
4	0*	3	1	2	4	4
5	1	7	1*	2	5	2
6	6	2	2	5	4	4
7	3	2	2	2	2	1*
8	3	2	3	5	1*	1
9	4	4	4	3	6	3

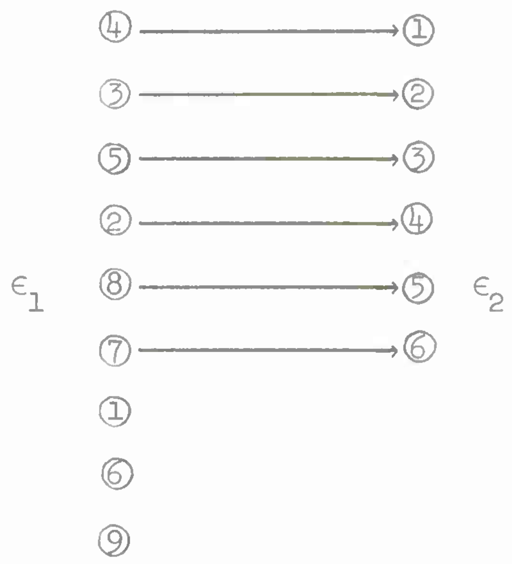
${}_ov^1 = 4$

; ${}_oC^2 =$

	1	2	3	4	5	6
1	2	0	1	4	0*	3
2	5	3	6	3	4	4
3	1*	4	2	1	5	2
4	6	3	5	4	2	3
5	7	0*	1	3	2	3
6	2	2	4	2	3	1*
7	4	6	4	2	5	3
8	2	9	2	1*	4	1
9	3	1	1*	2	2	2

${}_ov^2 = 4$

The optimal assignment for each C^k independently are shown by asterisks. Their values are v^k shown after the matrices. To show the details of the procedures described in the previous section consider C^1. The corresponding bipartite graph showing arcs (i,j) only, $i \in \epsilon$, $j \in \epsilon_2$ is:



Note therefore that the appropriate permutation is (4, 3, 5, 2, 8, 7, 1, 6, 9), or the permutation-matrix, P is

	1	2	3	4	5	6	7	8	9
1				1					
2			1						
3					1				
4		1							
5								1	
6							1		
7	1								
8						1			
9									1

Matrix W^1 is easily calculated as:

	1	2	3	4	5	6	-	-	-
4	0	2	0	2	3	3	0	0	0
3	4	0	1	9	0	1	0	0	0
5	1	6	0	2	4	1	0	0	0
2	0	3	0	0	2	1	0	0	0
8	3	1	2	5	0	0	0	0	0
7	3	1	1	2	1	0	0	0	0
1	4	5	4	3	6	2	0	∞	∞
6	6	1	1	5	3	3	∞	0	∞
9	4	3	3	3	5	2	∞	∞	0

The index-numbers are of the original C^1-matrix to clarify the procedure. Floyd's algorithm yields the following W^*:

	1	2	3	4	5	6	-	-	-
4	0	1	0	2	1	1	0	0	0
3	2	0	1	2	0	0	0	0	0
5	1	1	0	2	1	1	0	0	0
2	0	1	0	0	1	1	0	0	0
8	2	1	1	2	0	0	0	0	0
7	2	1	1	2	1	0	0	0	0
1	3	3	3	3	3	2	0	2	0
6	2	1	1	3	1	1	1	0	1
9	3	3	3	3	3	2	2	2	0

,

so that $W^1 + W^{1*t}$ becomes

	1	2	3	4	5	6	-	-	-
4	0	4	1	2	5	5	3	2	3
3	5	0	2	10	1	2	3	1	3
5	1	7	0	2	5	2	3	1	3
2	2	5	2	0	4	3	3	3	3
8	4	1	3	6	0	1	3	1	3
7	4	1	2	3	1	0	2	1	2
1	4	5	4	3	6	2	0	∞	∞
6	6	1	1	5	3	3	∞	0	∞
9	4	3	3	3	5	2	∞	∞	0

The relevant part, Δf^1, with Δf^2 (calculated in the same way) are displayed to the DM:

$$
{}_o\Delta f^1 = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7\\8\\9 \end{array}
\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\left[\begin{array}{cccccc}
4 & 5 & 4 & 3 & 6 & 2\\
2 & 5 & 2 & 0 & 4 & 3\\
5 & 0 & 2 & 10 & 1 & 2\\
0 & 4 & 1 & 2 & 5 & 5\\
1 & 7 & 0 & 2 & 5 & 2\\
6 & 1 & 1 & 5 & 3 & 3\\
4 & 1 & 2 & 3 & 1 & 0\\
4 & 1 & 3 & 6 & 0 & 1\\
4 & 3 & 3 & 3 & 5 & 2
\end{array}\right] \\ {}_o\ell f^1 = 4 \end{array}
\qquad
{}_o\Delta f^2 = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7\\8\\9 \end{array}
\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\left[\begin{array}{cccccc}
3 & 2 & 2 & 5 & 0 & 4\\
4 & 3 & 5 & 2 & 4 & 3\\
0 & 6 & 3 & 1 & 6 & 2\\
5 & 3 & 4 & 3 & 2 & 2\\
8 & 0 & 1 & 4 & 2 & 4\\
1 & 3 & 4 & 1 & 4 & 0\\
3 & 6 & 3 & 1 & 5 & 2\\
1 & 10 & 2 & 0 & 5 & 1\\
3 & 1 & 0 & 2 & 2 & 3
\end{array}\right] \\ {}_o\ell f^2 = 4 \end{array}
$$

Assume that the choices of the DM were:

j	1	2	3	4	5	6
$i_1(j)$	2	6	5	2	8	7
$i_2(j)$	8	9	4	4	7	8

$j(0) = 3$, so that the first assignment is $(5, 3)$ with $c^1_{5,3} = 1$, $c^2_{5,3} = 1$. The remaining C-matrices are:

$$
{}_1C^1 = \begin{array}{c|ccccc|}
 & 1 & 2 & 4 & 5 & 6 \\
\hline
1 & 4 & 6 & 3 & 7 & 3 \\
2 & 0 & 4 & 0^* & 3 & 2 \\
3 & 4 & 1^* & 9 & 1 & 2 \\
4 & 0^* & 3 & 2 & 4 & 4 \\
6 & 6 & 2 & 5 & 4 & 4 \\
7 & 3 & 2 & 2 & 2 & 1^* \\
8 & 3 & 2 & 5 & 1^* & 1 \\
9 & 4 & 4 & 3 & 6 & 3 \\
\hline
\end{array}
\qquad
{}_1C^2 = \begin{array}{c|ccccc|}
 & 1 & 2 & 4 & 5 & 6 \\
\hline
1 & 2 & 0 & 4 & 0^* & 3 \\
2 & 5 & 3 & 3 & 4 & 4 \\
3 & 1^* & 4 & 1 & 5 & 2 \\
4 & 6 & 3 & 4 & 2 & 3 \\
6 & 2 & 2 & 2 & 3 & 1^* \\
7 & 4 & 6 & 2 & 5 & 3 \\
8 & 2 & 9 & 1^* & 4 & 1 \\
9 & 3 & 1^* & 2 & 2 & 2 \\
\hline
\end{array}
$$

$${}_1v^1 = 3 \qquad {}_1v^2 = 4$$

from which we have:

$$
{}_1\Delta f^1 = \begin{array}{c|ccccc|}
 & 1 & 2 & 4 & 5 & 6 \\
\hline
1 & 4 & 5 & 3 & 6 & 2 \\
2 & 2 & 5 & 0 & 4 & 3 \\
3 & 5 & 0 & 10 & 1 & 2 \\
4 & 0 & 4 & 2 & 5 & 5 \\
6 & 6 & 1 & 5 & 3 & 3 \\
7 & 4 & 1 & 3 & 1 & 0 \\
8 & 4 & 1 & 6 & 0 & 1 \\
9 & 4 & 3 & 3 & 5 & 2 \\
\hline
\end{array}
\qquad
{}_1\Delta f^2 = \begin{array}{c|ccccc|}
 & 1 & 2 & 4 & 5 & 6 \\
\hline
1 & 3 & 1 & 5 & 0 & 4 \\
2 & 4 & 2 & 2 & 4 & 3 \\
3 & 0 & 5 & 1 & 6 & 2 \\
4 & 5 & 2 & 3 & 2 & 2 \\
6 & 1 & 2 & 1 & 3 & 0 \\
7 & 3 & 5 & 1 & 5 & 2 \\
8 & 1 & 9 & 0 & 4 & 1 \\
9 & 3 & 0 & 2 & 1 & 2 \\
\hline
\end{array}
$$

$${}_1\ell f^1 = 4 \qquad {}_1\ell f^2 = 5$$

Note that in the calculation of ${}_1\Delta f^2$ a negative element $(c^2_{1,2} - c^2_{9,2} = -1)$ was involved, still no difficulties arose as predicted by the proof of the theorem.

The choices of the DM were:

j	1	2	4	5	6
$i_1(j)$	4	6	2	8	7
$i_2(j)$	8	9	4	9	8

$j(1)$ is 4 so that the second assignment is (2,4) with $c^1_{2,4} = 0$, $c^2_{2,4} = 3$ and the remaining C-matrices are:

$$
{}_2C^1 = \begin{array}{c|cccc|} & 1 & 2 & 5 & 6 \\ \hline 1 & 4 & 6 & 7 & 3 \\ 3 & 4 & 1^* & 1 & 2 \\ 4 & 0^* & 3 & 4 & 4 \\ 6 & 6 & 2 & 4 & 4 \\ 7 & 3 & 2 & 2 & 1^* \\ 8 & 3 & 2 & 1^* & 1 \\ 9 & 4 & 4 & 6 & 3 \\ \hline \end{array}
\qquad
{}_2C^2 = \begin{array}{c|cccc|} & 1 & 2 & 5 & 6 \\ \hline 1 & 2 & 0 & 0^* & 3 \\ 3 & 1^* & 4 & 5 & 2 \\ 4 & 6 & 3 & 2 & 3 \\ 6 & 2 & 2 & 3 & 1^* \\ 7 & 4 & 6 & 5 & 3 \\ 8 & 2 & 9 & 4 & 1 \\ 9 & 3 & 1^* & 2 & 2 \\ \hline \end{array}
$$

$${}_2v^1 = 3 \qquad\qquad {}_2v^2 = 3$$

from which:

$$
{}_2\Delta f^1 = \begin{array}{c|cccc|}
 & 1 & 2 & 5 & 6 \\
\hline
1 & 4 & 5 & 6 & 2 \\
3 & 5 & 0 & 1 & 2 \\
4 & 0 & 5 & 6 & 6 \\
6 & 6 & 1 & 3 & 3 \\
7 & 4 & 1 & 1 & 0 \\
8 & 4 & 1 & 0 & 1 \\
9 & 4 & 3 & 5 & 2 \\
\hline
\end{array}
\qquad
{}_2\Delta f^2 = \begin{array}{c|cccc|}
 & 1 & 2 & 3 & 4 \\
\hline
1 & 3 & 1 & 0 & 2 \\
3 & 0 & 4 & 5 & 2 \\
4 & 5 & 2 & 2 & 2 \\
6 & 1 & 1 & 2 & 0 \\
7 & 3 & 5 & 5 & 2 \\
8 & 1 & 8 & 4 & 0 \\
9 & 3 & 0 & 1 & 2 \\
\hline
\end{array}
$$

$$ {}_2\ell f^1 = 4 \qquad {}_2\ell f^2 = 7 $$

The choices of the DM were:

j	1	2	5	6
$i_1(j)$	8	9	9	8
$i_2(j)$	6	6	6	6

$j(2) = 6$, the third assignment is $(8,6)$: $c^1_{8,6} = 1$, $c^2_{8,6} = 1$. The remaining C-matrices are:

$$
{}_3C^1 = \begin{array}{c|ccc|}
 & 1 & 2 & 5 \\
\hline
1 & 4 & 6 & 7 \\
3 & 4 & 1^* & 1 \\
4 & 0^* & 3 & 4 \\
6 & 6 & 2 & 4 \\
7 & 3 & 2 & 2^* \\
9 & 4 & 4 & 6 \\
\hline
\end{array}
\qquad
{}_3C^2 = \begin{array}{c|ccc|}
 & 1 & 2 & 5 \\
\hline
1 & 2 & 0 & 0^* \\
3 & 1^* & 4 & 5 \\
4 & 6 & 3 & 2 \\
6 & 2 & 2 & 3 \\
7 & 4 & 6 & 5 \\
9 & 3 & 1^* & 2 \\
\hline
\end{array}
$$

$$ {}_3v^1 = 3 \qquad {}_3v^2 = 2 $$

$$
{}_3\Delta f^2 = \begin{array}{c|ccc|}
 & 1 & 2 & 5 \\ \hline
1 & 4 & 4 & 5 \\
3 & 5 & 0 & 0 \\
4 & 0 & 4 & 5 \\
6 & 6 & 0 & 2 \\
7 & 3 & 0 & 0 \\
9 & 4 & 2 & 4 \\ \hline
\end{array}
\qquad
{}_3\Delta f^2 = \begin{array}{c|ccc|}
 & 1 & 2 & 5 \\ \hline
1 & 3 & 1 & 0 \\
3 & 0 & 4 & 5 \\
4 & 5 & 2 & 2 \\
6 & 1 & 1 & 3 \\
7 & 3 & 5 & 5 \\
9 & 3 & 0 & 1 \\ \hline
\end{array}
$$

$$
{}_3\ell f^1 = 5 \qquad\qquad {}_3\ell f^2 = 7
$$

The choices of the DM were:

$$
\begin{array}{c|ccc|}
\hline
j & 1 & 2 & 5 \\
i_1(j) & 3 & 6 & 9 \\
i_2(j) & 7 & 9 & 6 \\ \hline
\end{array}
$$

$j(3) = 2$, the fourth assignment: $(6,2)$: $c^1_{6,2} = 2$, $c^2_{6,2} = 2$, leaving:

$$
{}_4C^1 = \begin{array}{c|cc|}
 & 1 & 5 \\ \hline
1 & 4 & 7 \\
3 & 4 & 1^* \\
4 & 0^* & 4 \\
7 & 3 & 2 \\
9 & 4 & 6 \\ \hline
\end{array}
\qquad
{}_4C^2 = \begin{array}{c|cc|}
 & 1 & 5 \\ \hline
1 & 2 & 0^* \\
3 & 1^* & 5 \\
4 & 6 & 2 \\
7 & 4 & 5 \\
9 & 3 & 2 \\ \hline
\end{array}
$$

$$
{}_4v^1 = 1 \qquad\qquad {}_4v^2 = 1
$$

$$
{}_4\Delta f^1 = \begin{array}{c|cc|}
 & 1 & 5 \\
\hline
1 & 4 & 6 \\
3 & 5 & 0 \\
4 & 0 & 6 \\
7 & 3 & 1 \\
9 & 4 & 5 \\
\hline
\end{array}
\qquad
{}_4\Delta f^2 = \begin{array}{c|cc|}
 & 1 & 5 \\
\hline
1 & 3 & 0 \\
3 & 0 & 6 \\
4 & 5 & 2 \\
7 & 3 & 5 \\
9 & 2 & 2 \\
\hline
\end{array}
$$

$${}_4\ell f^1 = 5 \qquad {}_4\ell f^2 = 8$$

The choices of the DM:

j	1	5
$i_1(j)$	9	9
$i_2(j)$	7	1

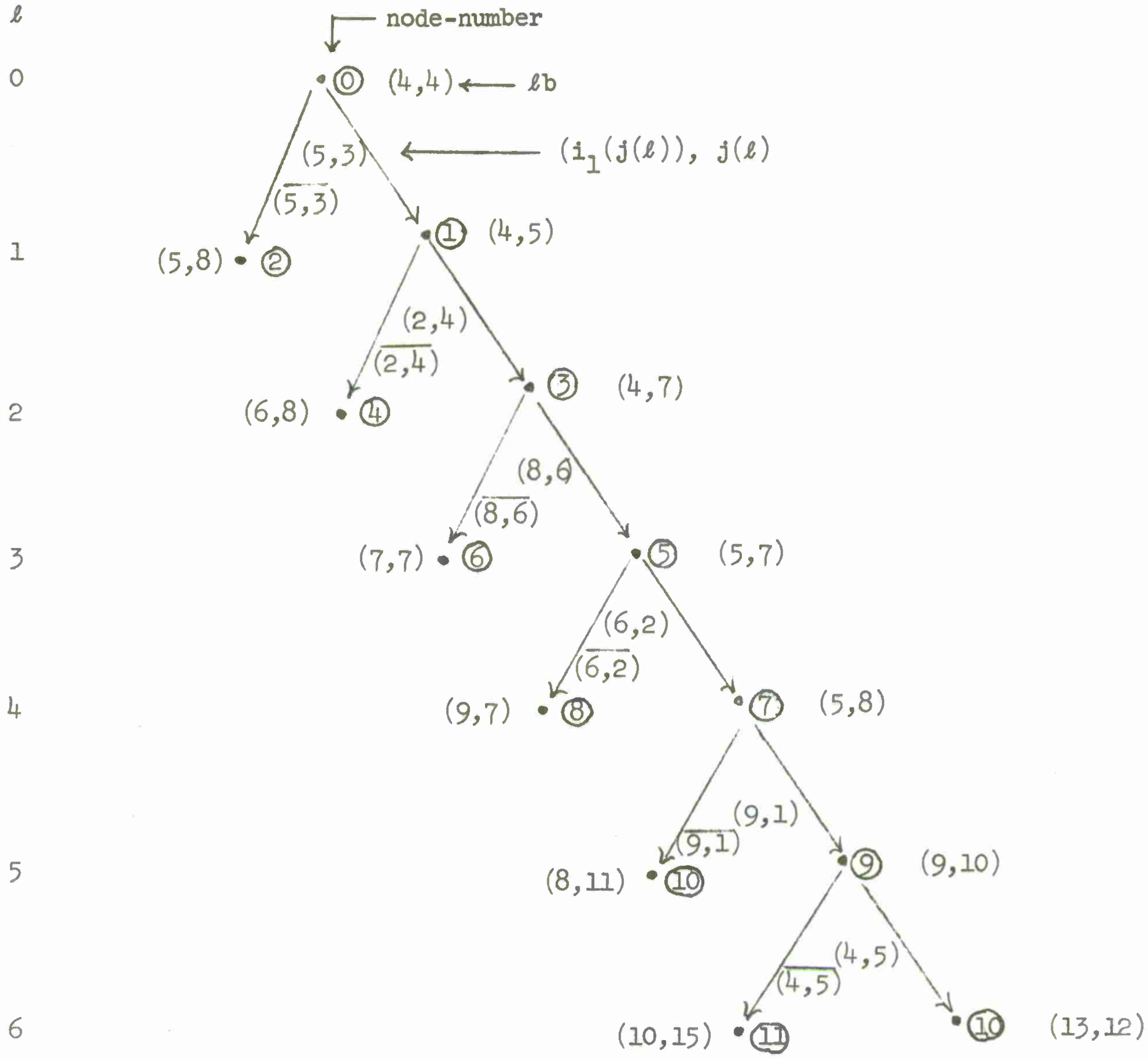
$j(4) = 1$, the fifth assignment: $(9,1)$: $c_{9,1} = 4$; $c_{9,1} = 3$; leaving:

$$
{}_5C^1 = \begin{array}{c|c|}
 & 5 \\
\hline
1 & 7 \\
3 & 1^* \\
4 & 4 \\
7 & 2 \\
\hline
\end{array}\,,
\quad
{}_5C^2 = \begin{array}{c|c|}
 & 5 \\
\hline
1 & 0^* \\
3 & 5 \\
4 & 2 \\
7 & 5 \\
\hline
\end{array}\,,
\quad
{}_5\Delta f^1 = \begin{array}{c|c|}
 & 5 \\
\hline
1 & 6 \\
3 & 0 \\
4 & 3 \\
7 & 1 \\
\hline
\end{array}\,,
\quad
{}_5\Delta f^2 = \begin{array}{c|c|}
 & 5 \\
\hline
1 & 0 \\
3 & 5 \\
4 & 2 \\
7 & 5 \\
\hline
\end{array}\,.
$$

$${}_5v^1 = 1 \qquad {}_5v^2 = 0 \qquad {}_5\ell f^1 = 9 \qquad {}_5\ell f^2 = 10$$

The choice of the DM $i_1(5) = 4$ and the last assignment is $(4, 5)$: $c^1_{4,5} = 4$, $c^2_{4,5} = 2$ and $f^1 = 13$, $f^2 = 12$.

At this stage the branch-and-bound tree with the lower bounds are displayed showing $(i_1(j(\ell)), j(\ell))$



Assume that the DM wants to explore the consequences of different assesment at node 7 (referring to previous ${}_4\Delta f^k$):

$$\begin{array}{r|cc|}
j & 1 & 5 \\
i_1(j) & 7 & 9 \\
i_2(j) & 1 & 1
\end{array},$$

$j(4) = 1$, the fifth assignment: $(7, 1)$ $c^1_{7,1} = 3$, $c^2_{7,1} = 4$,

$$_4c^1 = \begin{array}{r|c|}
 & 5 \\
1 & 7 \\
3 & 1* \\
4 & 4 \\
9 & 6
\end{array} \qquad _4c^1 = \begin{array}{r|c|}
 & 5 \\
1 & 0* \\
3 & 5 \\
4 & 2 \\
9 & 2
\end{array} \qquad _5\Delta f^2 = \begin{array}{|c|}
6 \\
0 \\
3 \\
5
\end{array} \qquad _5\Delta f^2 = \begin{array}{|c|}
0 \\
5 \\
2 \\
2
\end{array}$$

$$_5v^1 = 1 \qquad _5v^2 = 0 \qquad _5\ell f^1 = 8 \qquad _5\ell f^2 = 11$$

and the best assignment is $(4, 5)$, $c^1_{4,5} = 4$, $c^2_{4,5} = 2$, and $f^1 = 11$, $f^2 = 13$.

CONCLUSION

The derivation of Δf is original: note that it gives the exact increase of the objective-function if an edge is introduced to the optimal solution. This is in contradiction with [7], [8] where only an easily calculable lower bound of the increase is obtained. However, it was felt that the DM would like to have the possible tightest bounds which enable better decisions. This consideration outweighs the question of numerical efficiency. This is obviously the case if the time of DM is more expensive than the cost of calculation which appears to be the case in practical situations.

The calculations of the shortest routes are of the order $O(n^3)$, so that all the calculations of a single branch of the branch-and-bound tree (from $\ell = 0$ to $\ell = m$) is of the order $O(n^4)$. If a modest number of branches is investigated the calculations do not take a prohibitively long time or space. Here it can be mentioned that the usual Floyd algorithm may be more efficiently solved by techniques discussed in [1].

Thus the proposed method is computationally feasible, while from the point of view of methodology it is similar to that which have had success in the convex programming saituation [11], consequently it can be an applicable tool in personnel-assigning.

REFERENCES

1. Brucker, P., All Shortest Distances in Networks with Large Number of Strong Components, 41st National Meeting of ORSA, New Orleans, 1972.

2. Dantzig, G.B., Linear Programming and Extensions, Princeton University Press, Princeton (N.J.), 1963.

3. Dreyfus, S.E., An Appraisal of Some Shortest Path Algorithms, Oper. Res., 17, 395-412 (1969).

4. Ewoshko, T.A., and Dudding, R.C., Application of Kuhn's Hungarian Assignment Algorithm to Posting Servicemen, Oper. Res., 19, 991 (1971).

5. Floyd, R.W., Algorithm 97: Shortest Path, Communication of ACM, 6, 345- (1962).

6. Geoffrion, A.M., Vector Maximal Decomposition Programming, 7th Mathematical Programming Symposium, The Hague, September 1970.

7. Hillier, F. S. and Lieberman, G. J., Introduction to Operations Research, Holden-Day, 1967, San Francisco, pp. 172-204.

8. Little, J.C.C. et al., In Algorithm for the Traveling Salesman Problem, Oper. Res., 11, 972-989 (1963).

9. Machol, R.E., An Application of the Assignment Problem, Opers. Res. 18, 745-746 (1970).

10. Pollatschek, M.A., A Necessary and Sufficient Condition for the Existence of Hamiltonian Paths and Circuits and its Application to the Traveling-Salesman Problem, Mimeographed, Technion, Haifa, Israel, 1971.

11. Roy, B., Problems and Methods with Multiple Objective Functions, Mathematical Programming, 1, 239-266 (1971).